22C:44 Homework 7
Due November 2, 2000

Each problem is worth 10 points.

1. Problem 12-4 on page 242 from the textbook.
2. The *knapsack problem* is as follows: given n rods of length $s_1, s_2, \ldots, s_n$, and a natural number S, find a subset of the rods that has total length exactly S. Suppose that $s_1, s_2, \ldots, s_n$ are all natural numbers as well.

   Design a dynamic programming algorithm for this problem. Analyze the time complexity and space complexity of the algorithm.
   Hint: Check if there is a subset of the first $n-1$ rods whose total length adds up to S or $S - s_n$.
3. A certain string processing language allows the programmer to break a string into two pieces. Since this involves copying the old string, it costs n units of time to break a string of n characters into two pieces. Suppose a programmer wants to break a string into many pieces. The order in which the breaks are made can affect the total amount of time used. For example, suppose we wish to break a 20 character string after characters 3, 8, and 10 (numbering the characters in ascending order from left to right starting from 1). If the breaks are made in left-to-right order, then the first break costs 20 units of time, the second break costs 17 units of time, and the third break costs 12 units of time, for a total of 49 units of time. If the breaks are made in the right-to-left order, then the first break costs 20 units, the second break costs 10 units, and the third break costs 8 units of time, for a total of 38 units of time. Devise a dynamic programming algorithm that, when given the numbers of characters after which to break, determines the cheapest cost of those breaks in $O(n^3)$ time.
4. Problem 16-1 on page 324 in the textbook.